

# 一、protobuf是什么？

---

protobuf(Google Protocol Buffers)是Google提供一个具有高效的协议数据交换格式工具库(类似Json)，但相比于Json，Protobuf有更高的转化效率，时间效率和空间效率都是JSON的3-5倍。

## 二、protobuf有什么用？

---

Xml、Json是目前常用的数据交换格式，它们直接使用字段名称维护序列化后类实例中字段与数据之间的映射关系，一般用字符串的形式保存在序列化后的字节流中。消息和消息的定义相对独立，可读性较好。但序列化后的数据字节很大，序列化和反序列化的时间较长，数据传输效率不高。

Protobuf和Xml、Json序列化的方式不同，采用了二进制字节的序列化方式，用字段索引和字段类型通过算法计算得到字段之前的关系映射，从而达到更高的时间效率和空间效率，特别适合对数据大小和传输速率比较敏感的场所使用。

## 三、Protobuf与json的对比

---

### 1、创建product.proto文件

---

```
message SearchRequest {
    required string phone_name = 1;
    optional int32 price = 2;
    optional int32 top = 3;
}
```

### 2、消息结构和java对象赋值

---

```
PhoneName:" idol3"
Price:2000
Top:1

WatchName:" tcl watch"
Price:1000
Top:1
```

### 3、JSON字符串

---

```
{"phone":{"phoneName":"idol3","price":2000,"top":1},"watch":{"watchName":"tcl
wtch","top":1,"price":1000}}
```

### 4、Protobuf转化后的二进制文件

---

优点：protobuf的空间效率是JSON的2-5倍，时间效率要高，对于数据大小敏感，传输效率高的模块可以采用protobuf库

缺点：消息结构可读性不高，序列化后的字节序列为二进制序列不能简单的分析有效性；目前使用不广泛，只支持java,C++和Python；

## 四、protobuf有什么？

Protobuf 提供了C++、java、python语言的支持，提供了windows(proto.exe)和linux平台动态编译生成proto文件对应的源文件。proto文件定义了协议数据中的实体结构(message ,field)

### 1. 关键字message:

代表了实体结构，由多个消息字段(field)组成。

### 2. 消息字段(field):

包括数据类型、字段名、字段规则、字段唯一标识、默认值

### 3. 数据类型:

常见的原子类型都支持

protobuf数据类型转换

.proto Type	Notes	C++ Type	Java Type	Python Type[2]	Go Type	Ruby Type	C# Type	PHP Type	Dart Type
double		double	double	float	float64	Float	double	float	double
float		float	float	float	float32	Float	float	float	double
int32	Uses variable-length encoding. Inefficient for encoding negative numbers – if your field is likely to have negative values, use sint32 instead.	int32	int	int	int32	Fixnum or Bignum (as required)	int	integer	int
int64	Uses variable-length encoding. Inefficient for encoding negative numbers – if your field is likely to have negative values, use sint64 instead.	int64	long	int/long[3]	int64	Bignum	long	integer/string[5]	Int64
uint32	Uses variable-length encoding.	uint32	int[1]	int/long[3]	uint32	Fixnum or Bignum (as required)	uint	integer	int

### 4. 字段规则:

required：必须初始化字段，如果没有赋值，在数据序列化时会抛出异常

optional：可选字段，可以不必初始化。

repeated：数据可以重复(相当于java 中的Array或List)

字段唯一标识：序列化和反序列化将会使用到。

**默认值**：在定义消息字段时可以给出默认值。